Лабораторная работа 1.

Черновик 0.6

Целью лабораторной работы является знакомство студентов с инструментарием, который используется для создания, отладки и сопровождения программ, написанных на языке программирования Си.

Студенты должны получить и закрепить на практике следующие знания и умения:

1. Изучить стадии компиляции программы, научиться компилировать и компоновать программу в командной строке (однофайловый проект).
2. Получить представление об организации объектных и исполняемых файлов, научиться анализировать информацию, которая в них содержится.
3. Познакомиться с интегрированной средой разработки *Qt Creator*. Научиться
   - создавать проекты в *Qt Creator*;
   - настраивать сборки проектов (release и debug);
   - отлаживать программы в этой среде.

Общее задание

1. Результат выполнения каждого пункта задания необходимо отразить в отчете. Для написания отчета используется wiki в GitLab.

2. Установите дома необходимое программное обеспечение. Если в процессе установки или настройки программного обеспечения вы столкнулись с какими-нибудь проблемами, опишите эти проблемы и способы, с помощью которых вы их преодолели.

3. Познакомьтесь со стадиями компиляции и результатами работы компилятора.
   a. Напишите небольшую программу (не "hello, world!"; обязательно должны присутствовать комментарии, директива define; в качестве такой задачи можно взять задачу 1 из индивидуального задания).
   b. Изучите стадии компиляции программы. В отчете приведите команду, которую необходимо написать, чтобы выполнилась очередная стадия компиляция, а также наиболее существенные результаты работы компилятора (см. презентацию лекции #1, слайды 11-17).
      Подсказка:
      Чтобы найти параметры, с которыми запускается компоновщик ld необходимо внимательно изучить выдачу команды

      ```
      gcc -v foo.c -Wl,--verbose
      ```

      При этом большую часть этой выдачи можно «выкинуть».
   c. Какие динамические библиотеки использует ваша программа (воспользуйтесь утилитой *objdump* или *ldd*)?

4. Познакомьтесь с сообщениями, которые выдает компилятор во время компиляции программы с синтаксическими ошибками.
   a. Скомпилируйте программу, которая содержит синтаксические ошибки (задача 2 индивидуального задания).

b. Изучите выдачу компилятора. Какая информация помогает вам понять к какой строке исходной программы относится сообщение об ошибке?
c. Исправьте ошибки. Ошибки исправляются строго по одной. В отчет выносятся пары: сообщение компилятора об ошибке и соответствующие исправление. Можно посоветовать следующую последовательность действий: компиляция; копирование сообщения о первой ошибке в отчет; исправление ошибки; использование команды git diff, чтобы показать суть исправлений; использование команды git commit для фиксации изменений.

5. Познакомьтесь с интегрированной средой разработки *QT Creator*.
   В результате знакомства вы должны уметь:
   a. Создавать проект, настраивать этапы сборки и очистки проекта, запуск приложения.
   b. Уметь анализировать сообщения об ошибках компиляции; знать, где посмотреть сообщения об ошибках, выданные самим компилятором.
   c. Уметь использовать отладчик: выполнять программу в пошаговом режиме; устанавливать точки останова и условия их срабатывания; работать с переменными.

6. Выполните индивидуальное задание согласно варианту.
   a. Исходный код лабораторной работы располагается в ветке lab_01, а каждая из задач в отдельной папке: lab_01_X_1, lab_01_X_2, lab_01_X_3, lab_01_X_4, где вместо X указывается номер варианта (например, если у вас второй вариант, то папка будет называться lab_01_2_1 и т.д.).
   b. Исходный код должен соответствовать правилам оформления исходного кода.
   c. Для каждой задачи создается отдельный проект в *QT Creator*. Для каждого проекта должно быть два варианта сборки: Debug (с отладочной информацией) и Release (без отладочной информации).
   d. Для каждой задачи студентом подготавливаются тестовые данные, которые демонстрируют правильность ее работы. Эти данные в формате (как входные, так и результат) должны располагаться в файле test.txt.
   e. Для реализации задач этой лабораторной работы вам понадобится только функция main.
   f. Исходный код лабораторной работы помещается под версионный контроль. После того, как разработка закончена, в GitLab подготавливается merge request для переноса изменений из ветки develop в ветку master.

Индивидуальное задание

`Номер задания = Номер в журнале % Количество вариантов.`

Задача 1

0. Даны основания и высота равнобедренной трапеции. Найти периметр трапеции.
1. Даны основания равнобедренной трапеции и угол при большем основании. Найти площадь трапеции.
2. Треугольник задан координатами своих вершин. Найти периметр треугольника.

Задача 2 (один вариант для всех)

```
include studio.h
main{}
(
    int s;

    s: = 56;
    print (Year has s weeks)
)
```

Задача 3

0. Определить нормальный вес человека и индекс массы его тела по формулам: h * t / 240 и m / h^2, где h – рост человека (измеряемый в сантиметрах в первой формуле и в метрах – во второй); t – длина окружности грудной клетки (в сантиметрах); m – вес (в килограммах).
1. Смешано v1 литров воды температуры t1 с v2 литрами воды температуры t2. Найти объем и температуры образовавшейся смеси.
2. Три сопротивления R1, R2, R3 соединены параллельно. Найти сопротивление соединения.

Задача 4

0. Бутылка воды стоит 45 копеек. Пустые бутылки сдаются по 20 копеек, и на полученные деньги опять покупается вода. Какое наибольшее количество бутылок воды можно купить, имея некоторую сумму денег S копеек.
1. Определите номер подъезда и этажа по номеру квартиры девятиэтажного дома, считая, что на каждом этаже ровно 4 квартиры, а нумерация квартир начинается с первого подъезда.
2. Задано время в секундах. Перевести в часы, минуты, секунды.